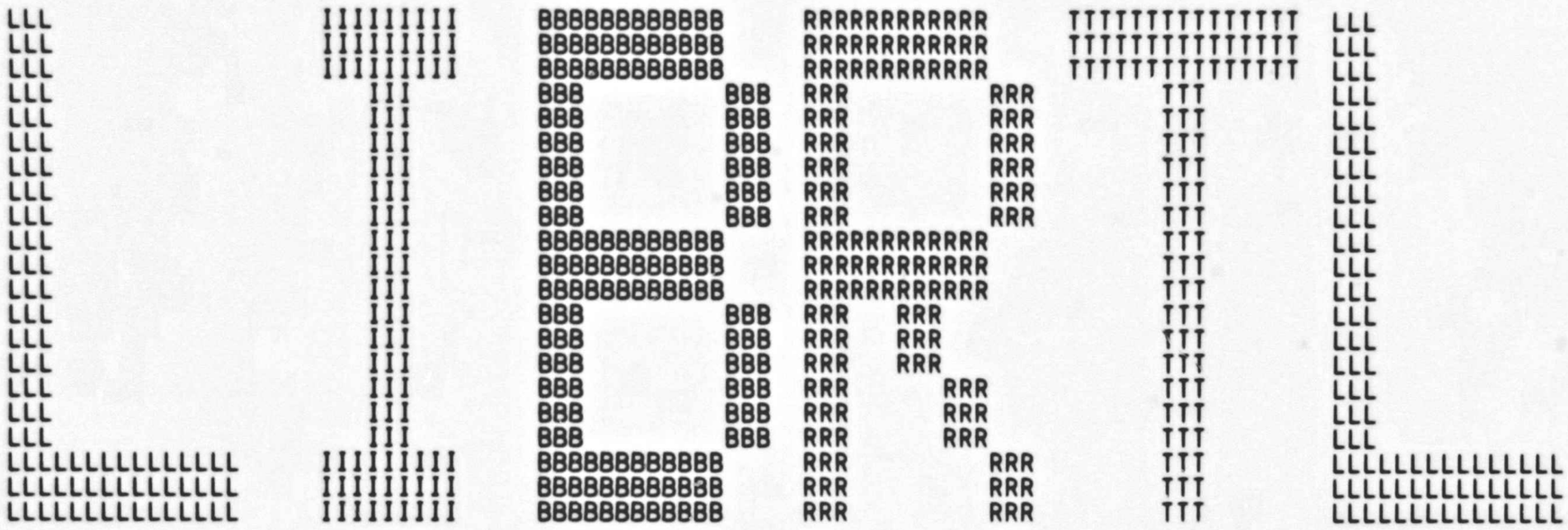
LLL IIIIIIIII BBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL IIIIIIIII BBBBBBBBBBBB RRRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL IIIIIIIII BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBBBBBBBBBB RRRRRRRRRRR TTT LLL
LLL III BBBBBBBBBBBB RRRRRRRRRRRR TTT LLL
LLL III BBBBBBBBBBBB RRRRRRRRRRRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**STRUPCASE

```
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR    UU      UU  PPPPPPPP     CCCCCCCC    AAAAAA     SSSSSSSS  EEEEEEEEEE
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR    UU      UU  PPPPPPPP     CCCCCCCC    AAAAAA     SSSSSSSS  EEEEEEEEEE
SS              TT      RR      RR  UU      UU  PP      PP  CC          AA      AA  SS        EE
SS              TT      RR      RR  UU      UU  PP      PP  CC          AA      AA  SS        EE
SS              TT      RR      RR  UU      UU  PP      PP  CC          AA      AA  SS        EE
SS              TT      RR      RR  UU      UU  PP      PP  CC          AA      AA  SS        EE
  SSSSSS        TT      RRRRRRRR    UU      UU  PPPPPPPP    CC          AA      AA    SSSSSS  EEEEEEEE
  SSSSSS        TT      RRRRRRRR    UU      UU  PPPPPPPP    CC          AA      AA    SSSSSS  EEEEEEEE
        SS      TT      RR  RR      UU      UU  PP          CC          AAAAAAAAAA          SS  EE
        SS      TT      RR  RR      UU      UU  PP          CC          AAAAAAAAAA          SS  EE
        SS      TT      RR    RR    UU      UU  PP          CC          AA      AA          SS  EE                ....
        SS      TT      RR    RR    UU      UU  PP          CC          AA      AA          SS  EE                ....
SSSSSSSS        TT      RR      RR  UUUUUUUUUU  PP            CCCCCCCC  AA      AA  SSSSSSSS  EEEEEEEEEE      ....
SSSSSSSS        TT      RR      RR  UUUUUUUUUU  PP            CCCCCCCC  AA      AA  SSSSSSSS  EEEEEEEEEE      ....
```

```
LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE STR$UPCASE (                  ! copy source to dest and upcase
;     2          0002  0
;     3          0003  0                     IDENT = '1-006' ! File: STRUPCASE.B32   Edit: DG1006
;     4          0004  0
;     5          0005  0                       ) =
;     6          0006  1 BEGIN
;     7          0007  1 !
;     8          0008  1 !**********************************************************************************
;     9          0009  1 !*                                                                                *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                         *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                           *
;    13          0013  1 !*                                                                                *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED         *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER        *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY        *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
;    19          0019  1 !*  TRANSFERRED.                                                                   *
;    20          0020  1 !*                                                                                *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE        *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT        *
;    23          0023  1 !*  CORPORATION.                                                                   *
;    24          0024  1 !*                                                                                *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS        *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                        *
;    27          0027  1 !*                                                                                *
;    28          0028  1 !*                                                                                *
;    29          0029  1 !**********************************************************************************
;    30          0030  1 !
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 ! FACILITY: String support library
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module takes strings of any supported class or dtype and
;    38          0038  1 !       copies the source string to the destination string, upcasing
;    39          0039  1 !       all letters a-z and all letters with diacritical marks.
;    40          0040  1 !       This routine is needed so that if a user wants any character
;    41          0041  1 !       comparison to be case blind, he will first convert both strings
;    42          0042  1 !       to upper case.
;    43          0043  1 !
;    44          0044  1 ! ENVIRONMENT: User mode, AST level or not or mixed
;    45          0045  1 !
;    46          0046  1 ! AUTHOR: R. Will, CREATION DATE: 1-Dec-79
;    47          0047  1 !
;    48          0048  1 ! MODIFIED BY:
;    49          0049  1 !
;    50          0050  1 !  R. Will, 1-Dec-79: VERSION 01
;    51          0051  1 ! 1-001 - Original
;    52          0052  1 ! 1-002 - String speedup, status from macros.  RW  11-Jan-1980
;    53          0053  1 ! 1-003 - Enhance to recognize additional classes of descriptors by
;    54          0054  1 !         using $STR$GET_LEN_ADDR to extract length and address of
;    55          0055  1 !         first data byte.  Remove string interlocking code.
;    56          0056  1 !         RKR 29-APR-1981
;    57          0057  1 ! 1-004 - Speed up code.   RKR 7-OCT-1981.
```

```
;    58          0058  1 ! 1-005 - Use STR$COPY_R_R8 for copying.  Use $STR$SIGNAL_FATAL
;    59          0059  1 !           instead of $STR$CHECK_STATUS.  RKR 18-NOV-1981.
;    60          0060  1 ! 1-006 - Enhance to use DEC Multinational Char. Set.  DG 1-Nov-1983.
;    61          0061  1 !--
;    62          0062  1
;    63          0063  1 !<BLF/PAGE>
```

```
;    65          0064  1 !
;    66          0065  1 ! SWITCHES:
;    67          0066  1 !
;    68          0067  1
;    69          0068  1 SWITCHES ADDRESSING_MODE
;    70          0069  1                 (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    71          0070  1
;    72          0071  1 !
;    73          0072  1 ! LINKAGES:
;    74          0073  1 !
;    75          0074  1
;    76          0075  1 REQUIRE 'RTLIN:STRLNK';         ! Use require file with string linkage
;    77          0260  1
;    78          0261  1 !
;    79          0262  1 ! TABLE OF CONTENTS:
;    80          0263  1 !
;    81          0264  1
;    82          0265  1 FORWARD ROUTINE
;    83          0266  1     STR$UPCASE;                 ! convert lowercase to uppercase
;    84          0267  1
;    85          0268  1 !
;    86          0269  1 ! INCLUDE FILES:
;    87          0270  1 !
;    88          0271  1
;    89          0272  1 REQUIRE 'RTLIN:RTLPSECT';       ! Declare PSECTS code
;    90          0367  1
;    91          0368  1 REQUIRE 'RTLIN:STRMACROS';      ! use string macros to code
;    92          1284  1
;    93          1285  1 LIBRARY 'RTLSTARLE';            ! STARLET library for macros and symbl
;    94          1286  1
;    95          1287  1 !
;    96          1288  1 ! MACROS: NONE
;    97          1289  1 !
;    98          1290  1
;    99          1291  1 !
;   100          1292  1 ! EQUATED SYMBOLS: NONE
;   101          1293  1 !
;   102          1294  1
;   103          1295  1 !
;   104          1296  1 ! PSECT DECLARATIONS
;   105          1297  1 !
;   106          1298  1
;   107          1299  1 DECLARE_PSECTS (STR);
;   108          1300  1
;   109          1301  1 !
;   110          1302  1 ! OWN STORAGE: NONE
;   111          1303  1 !
;   112          1304  1
;   113          1305  1 !
;   114          1306  1 ! EXTERNAL REFERENCES:
;   115          1307  1 !
;   116          1308  1
;   117          1309  1 EXTERNAL ROUTINE
;   118          1310  1     LIB$STOP,                                  ! signal errors
;   119          1311  1     STR$COPY_R_R8 : STR$JSB_COPY_R ;           ! Routine to do the copy
;   120          1312  1
;   121          1313  1 EXTERNAL LITERAL
```

```
;  122          1314  1     STR$_NORMAL;                    ! successful completion
```

```
;  124     1315  1 GLOBAL ROUTINE STR$UPCASE (                  ! upcase
;  125     1316  1
;  126     1317  1         DEST_DESC,                  ! Pointer to destination descriptor
;  127     1318  1         SRC_DESC                    ! Pointer to source descriptor
;  128     1319  1
;  129     1320  1                                 ) : =
;  130     1321  1
;  131     1322  1 !++
;  132     1323  1 ! FUNCTIONAL DESCRIPTION:
;  133     1324  1 !
;  134     1325  1 !       This routine copies the input string to the output string  and
;  135     1326  1 !       changes all lower case letters a-z plus letter with diacritcal
;  136     1327  1 !       marks to upper case
;  137     1328  1 !
;  138     1329  1 ! FORMAL PARAMETERS:
;  139     1330  1 !
;  140     1331  1 !       DEST_DESC.wt.dx         pointer to destination string descriptor
;  141     1332  1 !       SRC_DESC.rt.dx          pointer to source string descriptor
;  142     1333  1 !
;  143     1334  1 ! IMPLICIT INPUTS:
;  144     1335  1 !
;  145     1336  1 !       NONE
;  146     1337  1 !
;  147     1338  1 ! IMPLICIT OUTPUTS:
;  148     1339  1 !
;  149     1340  1 !       NONE
;  150     1341  1 !
;  151     1342  1 ! COMPLETION CODES:
;  152     1343  1 !
;  153     1344  1 !       SS$_NORMAL      Success
;  154     1345  1 !       STR$_TRU        Truncation occurred.  Warning.
;  155     1346  1 !
;  156     1347  1 ! SIDE EFFECTS:
;  157     1348  1 !
;  158     1349  1 !       May signal:
;  159     1350  1 !           STR$_FATINTERR      Fatal internal error
;  160     1351  1 !           STR$_ILLSTRCLA      Illegal (or unsupported) string class
;  161     1352  1 !           STR$_INSVIRMEM      Insufficient virtual memory for
;  162     1353  1 !                               reallocation of dynamic string
;  163     1354  1 !
;  164     1355  1 !--
;  165     1356  1
;  166     1357  2     BEGIN
;  167     1358  2
;  168     1359  2     LOCAL
;  169     1360  2         IN_LEN,                         ! length of input string
;  170     1361  2         IN_ADDR,                        ! addr of 1st byte of input str.
;  171     1362  2         OUT_LEN,                        ! length of output string
;  172     1363  2                                         ! (after copy operation)
;  173     1364  2         OUT_ADDR,                       ! addr of 1st byte of output
;  174     1365  2                                         ! string (after copy operation)
;  175     1366  2         RETURN_STATUS;                  ! status from STR$COPY_R_R8
;  176     1367  2
;  177     1368  2     MAP
;  178     1369  2         SRC_DESC  : REF $STR$DESCRIPTOR,
;  179     1370  2         DEST_DESC : REF $STR$DESCRIPTOR;
;  180     1371  2
```

```
;  181     1372  2 !+
;  182     1373  2 ! Compute the length and addr of 1st byte of input string and signal
;  183     1374  2 ! if fatal error returned.
;  184     1375  2 !-
;  185     1376  2     $STR$GET_LEN_ADDR ( SRC_DESC, IN_LEN, IN_ADDR ) ;
;  186     1377  2
;  187     1378  2 !+
;  188     1379  2 ! Copy the string and signal if fatal error results.
;  189     1380  2 !-
;  190     1381  2
;  191     1382  2     RETURN_STATUS = STR$COPY_R_R8 ( .DEST_DESC, .IN_LEN, .IN_ADDR) ;
;  192     1383  2     $STR$SIGNAL_FATAL (RETURN_STATUS) ;
;  193     1384  2
;  194     1385  2 !+
;  195     1386  2 ! Compute the current length of the output string.
;  196     1387  2 !-
;  197     1388  2     $STR$GET_LEN_ADDR ( DEST_DESC, OUT_LEN, OUT_ADDR ) ;
;  198     1389  2
;  199     1390  2 !+
;  200     1391  2 ! The copy is finished, now upcase the destination string.
;  201     1392  2 !-
;  202     1393  2
;  203     1394  2     CH$TRANSLATE (                      ! ASCII translation table
;  204     1395  2         CH$TRANSTABLE (                 ! upcase the string
;  205     1396  2             0,  1,  2,  3,  4,  5,  6,  7,  8,  9,     ! translate to self
;  206     1397  2             10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
;  207     1398  2             20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
;  208     1399  2             30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
;  209     1400  2             40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
;  210     1401  2             50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
;  211     1402  2             60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
;  212     1403  2             70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
;  213     1404  2             80, 81, 82, 83, 84, 85, 86, 87, 88, 89,
;  214     1405  2             90, 91, 92, 93, 94, 95, 96,
;  215     1406  2
;  216     1407  2                                         65, 66, 67,  ! lower to upper
;  217     1408  2             68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
;  218     1409  2             78, 79, 80, 81, 82, 83, 84, 85, 86, 87,
;  219     1410  2             88, 89, 90,
;  220     1411  2
;  221     1412  2                                     123, 124, 125, 126, 127, 128, 129,  ! translate to self
;  222     1413  2             130, 131, 132, 133, 134, 135, 136, 137, 138, 139,
;  223     1414  2             140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
;  224     1415  2             150, 151, 152, 153, 154, 155, 156, 157, 158, 159,
;  225     1416  2             160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
;  226     1417  2             170, 171, 172, 173, 174, 175, 176, 177, 178, 179,
;  227     1418  2             180, 181, 182, 183, 184, 185, 186, 187, 188, 189,
;  228     1419  2             190, 191, 192, 193, 194, 195, 196, 197, 198, 199,
;  229     1420  2             200, 201, 202, 203, 204, 205, 206, 207, 208, 209,
;  230     1421  2             210, 211, 212, 213, 214, 215, 216, 217, 218, 219,
;  231     1422  2             220, 221, 222, 223,
;  232     1423  2
;  233     1424  2                                 192, 193, 194, 195, 196, 197,  ! lower to upper
;  234     1425  2             198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
;  235     1426  2
;  236     1427  2             240,                                               ! translate to self
;  237     1428  2
```

```
;  238          1429  2              209, 210, 211, 212, 213, 214, 215, 216, 217, ! lower to upper
;  239          1430  2          218, 219, 220, 221,
;  240          1431  2
;  241          1432  2                                  254, 255),                  ! translate to self
;  242          1433  2          .OUT_LEN,                                          ! from destination
;  243          1434  2          .OUT_ADDR,                                         !
;  244          1435  2          STR$K_FILL_CHAR,                                   ! never used since same
;  245          1436  2                                                             ! string
;  246          1437  2          .OUT_LEN,                                          ! to destination
;  247          1438  2          .OUT_ADDR);
;  248          1439  2
;  249          1440  2      RETURN .RETURN_STATUS;
;  250          1441  1      END;                                                   !End of STR$UPCASE

                                                        .TITLE  STR$UPCASE
                                                        .IDENT  \1-006\

                                                        .PSECT  _STR$CODE,NOWRT,  SHR,  PIC,2

0E  0D  0C  0B  0A  09  08  07  06  05  04  03  02  01  00  00000 P.AAA:  .BYTE   0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, - ;
1D  1C  1B  1A  19  18  17  16  15  14  13  12  11  10  0F  0000F                 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, -   ;
2C  2B  2A  29  28  27  26  25  24  23  22  21  20  1F  1E  0001E                 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, -   ;
3B  3A  39  38  37  36  35  34  33  32  31  30  2F  2E  2D  0002D                 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, -   ;
4A  49  48  47  46  45  44  43  42  41  40  3F  3E  3D  3C  0003C                 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, -   ;
59  58  57  56  55  54  53  52  51  50  4F  4E  4D  4C  4B  0004B                 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, -   ;
48  47  46  45  44  43  42  41  60  5F  5E  5D  5C  5B  5A  0005A                 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, -   ;
57  56  55  54  53  52  51  50  4F  4E  4D  4C  4B  4A  49  00069                 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, -   ;
86  85  84  83  82  81  80  7F  7E  7D  7C  7B  5A  59  58  00078                 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, -   ;
95  94  93  92  91  90  8F  8E  8D  8C  8B  8A  89  88  87  00087                 93, 94, 95, 96, 65, 66, 67, 68, 69, 70, -   ;
A4  A3  A2  A1  A0  9F  9E  9D  9C  9B  9A  99  98  97  96  00096                 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, -   ;
B3  B2  B1  B0  AF  AE  AD  AC  AB  AA  A9  A8  A7  A6  A5  000A5                 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, -   ;
C2  C1  C0  BF  BE  BD  BC  BB  BA  B9  B8  B7  B6  B5  B4  000B4                 123, 124, 125, 126, 127, -128, -127, -      ;
D1  D0  CF  CE  CD  CC  CB  CA  C9  C8  C7  C6  C5  C4  C3  000C3                 -126, -125, -124, -123, -122, -121, -120, - ;
C0  DF  DE  DD  DC  DB  DA  D9  D8  D7  D6  D5  D4  D3  D2  000D2                 -119, -118, -117, -116, -115, -114, -113, - ;
CF  CE  CD  CC  CB  CA  C9  C8  C7  C6  C5  C4  C3  C2  C1  000E1                 -112, -111, -110, -109, -108, -107, -106, - ;
FE  DD  DC  DB  DA  D9  D8  D7  D6  D5  D4  D3  D2  D1  F0  000F0                 -105, -104, -103, -102, -101, -100, -99, -  ;
                                                    FF  000FF                   -98, -97, -96, -95, -94, -93, -92, -91, -   ;
                                                                                -90, -89, -88, -87, -86, -85, -84, -83, -   ;
                                                                                -82, -81, -80, -79, -78, -77, -76, -75, -   ;
                                                                                -74, -73, -72, -71, -70, -69, -68, -67, -   ;
                                                                                -66, -65, -64, -63, -62, -61, -60, -59, -   ;
                                                                                -58, -57, -56, -55, -54, -53, -52, -51, -   ;
                                                                                -50, -49, -48, -47, -46, -45, -44, -43, -   ;
                                                                                -42, -41, -40, -39, -38, -37, -36, -35, -   ;
                                                                                -34, -33, -64, -63, -62, -61, -60, -59, -   ;
                                                                                -58, -57, -56, -55, -54, -53, -52, -51, -   ;
                                                                                -50, -49, -16, -47, -46, -45, -44, -43, -   ;
                                                                                -42, -41, -40, -39, -38, -37, -36, -35, -   ;
                                                                                -2, -1                                      ;

                                                    .EXTRN  LIB$STOP, STR$COPY_R_R8
                                                    .EXTRN  STR$_NORMAL, STR$ANALYZE_SDESC_R1

                                        07FC 00000  .ENTRY  STR$UPCASE, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 1315
                                                    R10                                                 ;
                5A 00000000G   00   9E 00002        MOVAB   STR$ANALYZE_SDESC_R1, R10                   ;
```

```
                                50        08    AC  D0 00009          MOVL    SRC_DESC, R0                                    ; 1376
                                02        03    A0  91 0000D          CMPB    3(R0), #2                                       ;
                                                09  1A 00011          BGTRU   1$                                              ;
                                53              60  3C 00013          MOVZWL  (R0), IN_LEN                                    ;
                                52        04    A0  D0 00016          MOVL    4(R0), IN_ADDR                                  ;
                                                08  11 0001A          BRB     2$                                              ;
                                                6A  16 0001C 1$:      JSB     STR$ANALYZE_SDESC_R1                            ;
                                53              50  D0 0001E          MOVL    R0, R3                                          ;
                                52              51  D0 00021          MOVL    R1, R2                                          ;
                                59        04    AC  D0 00024 2$:      MOVL    DEST_DESC, R9                                   ; 1382
                                51              53  D0 00028          MOVL    IN_LEN, R1                                      ;
                                50              59  D0 0002B          MOVL    R9, R0                                          ;
                                     00000000G  00  16 0002E          JSB     STR$COPY_R_R8                                   ;
                                56              50  D0 00034          MOVL    R0, RETURN_STATUS                               ;
                                10              56  E8 00037          BLBS    RETURN_STATUS, 3$                               ; 1383
        04          56          03              00  ED 0003A          CMPZV   #0, #3, RETURN_STATUS, #4                       ;
                                                09  12 0003F          BNEQ    3$                                              ;
                                                56  DD 00041          PUSHL   RETURN_STATUS                                   ;
                         00000000G  00          01  FB 00043          CALLS   #1, LIB$STOP                                    ;
                                50        02    A9  91 0004A 3$:      CMPB    3(R9), #2                                       ; 1388
                                                09  1A 0004E          BGTRU   4$                                              ;
                                50              69  3C 00050          MOVZWL  (R9), OUT_LEN                                   ;
                                51        04    A9  D0 00053          MOVL    4(R9), OUT_ADDR                                 ;
                                                05  11 00057          BRB     5$                                              ;
                                50              59  D0 00059 4$:      MOVL    R9, R0                                          ;
                                                6A  16 0005C          JSB     STR$ANALYZE_SDESC_R1                            ;
    FE9B    CF          20          61          50  2E 0005E 5$:      MOVTC   OUT_LEN, (OUT_ADDR), #32, P.AAA, OUT_LEN, -    ; 1438
                                61              50      00065         (OUT_ADDR)                                              ;
                                50              56  D0 00067          MOVL    RETURN_STATUS, R0                               ; 1440
                                                04  0006A          RET                                                     ; 1441

; Routine Size:  107 bytes,    Routine Base:  _STR$CODE + 0100

;  251          1442  1
;  252          1443  1 END                                                    !End of module
;  253          1444  1
;  254          1445  0 ELUDOM



;                                   PSECT SUMMARY
;
;      Name                     Bytes                          Attributes
;
;  _STR$CODE                        363  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                                Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;      File                              Total    Loaded   Percent      Mapped      Time
```

```
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1          9776          6          0        581          00:00.7
;
;                                       COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:STRUPCASE/OBJ=OBJ$:STRUPCASE MSRC$:STRUPCASE/UPDATE=(ENH$:STRUPCASE
;       )

; Size:          107 code + 256 data bytes
; Run Time:          00:05.8
; Elapsed Time:      00:28.1
; Lines/CPU Min:     15078
; Lexemes/CPU-Min:  38973
; Memory Used:   90 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LINKER
LIS

STRPOSIT
LIS

STRUNWDEQ
LIS

STRPOSEXT
LIS

STRREPLAC
LIS

STRSRCHN
LIS

STRRIGHT
LIS

LINKER

STRTRIM
LIS

LINK
MAP

PREFIX
REQ

ISDSORT
LIS

STRUPCASE
LIS

STRTRANSL
LIS

DATBAS
MDL

TIRAUX
REQ

ISGENC
REQ

STRPREFIX
LIS

DATBAS
LIS